本快速入门介绍了如何将ASP.NET Core Identity 和 IdentityServer4一起使用。

# **新建认证服务器**

# **建立ASP.NET Identity新项目**

创建一个ASP.NET Core Web应用程序

然后选择Web应用程序（MVC）

然后点击“更改身份验证”按钮，选择“个人用户账户”

# **修改hosting**

不要忘记修改hosting以在端口5000上运行。这非常重要，这将关系到继续使用现有的客户端和API项

目。

# **添加IdentityServer组件**

添加IdentityServer4.AspNetIdentity NuGet包。

# **Scopes 和 Clients 配置**

将之前快速入门的配置类（在Config.cs中）复制到此新项目中。

对mvc客户端的修改

**new** Client

{

ClientId = "mvc",

ClientName = "MVC Client",

AllowedGrantTypes = GrantTypes.HybridAndClientCredentials,

RequireConsent = **false**,

ClientSecrets =

{

**new** Secret("secret".Sha256())

},

RedirectUris = { "http://localhost:5002/signin-oidc" },

PostLogoutRedirectUris = { "http://localhost:5002/signout-callback-oidc" },

AllowedScopes =

{

IdentityServerConstants.StandardScopes.OpenId,

IdentityServerConstants.StandardScopes.Profile,

"api1"

},

AllowOfflineAccess = **true**

}

# **配置IdentityServer**

**ConfigureServices**：

现在的解决方案下，我们用AddAspNetIdentity替换该扩展方法来使用ASP.NET Identity用户。

**public** **void** ConfigureServices(IServiceCollection services)

{

.......

services.AddIdentityServer()

.AddDeveloperSigningCredential()

.AddInMemoryPersistedGrants()

.AddInMemoryIdentityResources(Config.GetIdentityResources())

.AddInMemoryApiResources(Config.GetApiResources())

.AddInMemoryClients(Config.GetClients())

.AddAspNetIdentity<ApplicationUser>(); // AspNetIdentity

........

}

**Configure**

使用UseIdentityServer代替了对UseIdentity的调用

**public** **void** Configure(IApplicationBuilder app, IHostingEnvironment env)

{

.......

app.UseIdentityServer();

app.UseMvc(routes =>

{

routes.MapRoute(

name: "default",

template: "{controller=Home}/{action=Index}/{id?}");

});

........

}

# **创建用户数据库**

鉴于这是一个新的ASP.NET Identity项目，您将需要创建数据库

# **创建用户**

此时，您应该能够运行项目并在数据库中创建/注册用户。 启动应用程序，并从主页点击“Register”链接：

并在注册页面上创建一个新的用户帐户：

现在你有一个用户帐户，你应该可以登录，使用客户端，并调用API。

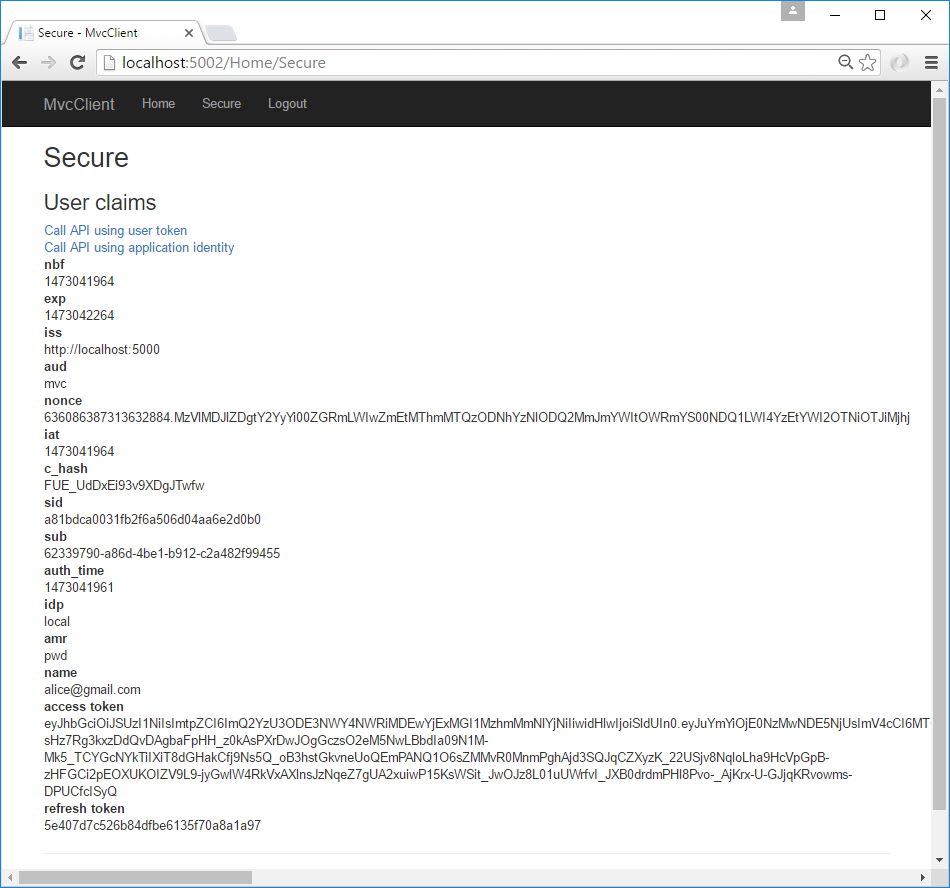
**客户端配置**

# **在MVC客户端登录**

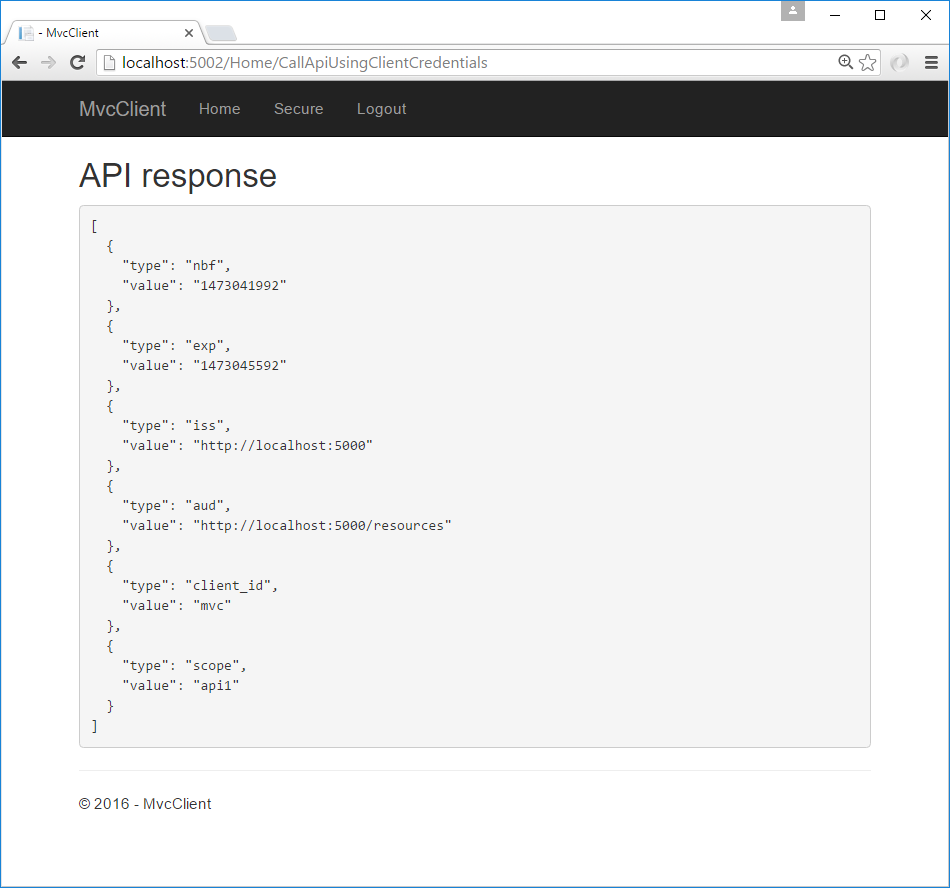
启动MVC客户端应用程序，你应该能够点击“Secure”链接登录。

您应该被重定向到ASP.NET Identity登录页面。 用新创建的用户登录：

登录后，您应该跳过同意页面（给出我们上面所做的更改），并立即重定向到MVC客户端应用程序，会显示你的用户信息。



您还应该能够单击“Call API using application identity”来调用API：



现在，您已经从ASP.NET Ientity的用户登录。